Docket No.: A0312.70410US00
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Stephen J. Plante et al.
Serial No.:	09/923,225
Confirmation No.:	8975
Filed:	August 6, 2001
For:	HIGH PERFORMANCE TURBO AND VITERBI CHANNEL DECODING IN DIGITAL SIGNAL PROCESSORS
Examiner:	J. D. Torres
Art Unit:	2112

Certificate of Mailing Under 37 CFR 1.8(a)
I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service on the date shown below with sufficient postage as First Class Mail, in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Dated: October 17, 2007

Doris A. Champagne

Doris A. Champagne

APPEAL BRIEF

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

As required under § 41.37(a), this brief is filed within two months of the Notice of Appeal filed in this case on August 2, 2007, and is in furtherance of said Notice of Appeal.

The fees required under § 41.20(b)(2) are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief contains items under the following headings as required by 37 C.F.R. § 41.37 and M.P.E.P. § 1205.2:

I. Real Party In Interest
II. Related Appeals and Interferences

I. REAL PARTY IN INTEREST

The real party in interest for this appeal is:

Analog Devices, Inc.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

A. Total Number of Claims in Application

There are 9 claims pending in application.

B. Current Status of Claims

1. Claims canceled: Claims 7-17, and 19-24.

2. Claims withdrawn from consideration but not canceled: None

3. Claims pending: Claims 1-6, 18, 25, 26.

4. Claims allowed: None

5. Claims rejected: Claims 1-6, 18, 25, 26.

C. Claims On Appeal

The claims on appeal are claims 1-6, 18, 25, 26.

IV. STATUS OF AMENDMENTS

Applicant did not file an amendment after the final Office Action mailed May 2, 2007. All prior amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The claimed subject matter relates generally to a method and a processor for processing signal values wherein an ACS (Add-Compare-Select) operation is performed in response to a single trellis instruction (page 10, line 32 to page 11, line 2). A block diagram of an embodiment of the processor is shown in FIGs. 7 and 8. Dataflow diagrams of operations performed by the processor in response to embodiments of the trellis instruction are shown in FIGs. 9 and 10. An example of software code that utilizes the trellis instruction to calculate alpha metrics and beta metrics of a trellis instruction is shown in FIG. 11. The claimed method and processor provide enhanced execution of turbo and Viterbi channel decoding algorithms typically used in wireless systems.

A block diagram of an embodiment of a digital signal processor (DSP) 110 suitable for execution of the trellis instruction is shown in FIG. 7. The DSP 110 includes computation blocks 112 and 114, a memory 116 and a control block 124 (page 7, lines 15-19). Control block 124 includes a program sequencer (FIG. 7). Memory 116 includes memory banks 140, 142 and 144 (page 7, lines 26-27).

A block diagram of an embodiment of each of the computation blocks 112 and 114 is shown in FIG. 8. The computation block includes a register file 200, a multiplier/accumulator

210, an arithmetic logic unit (ALU) 212, a shifter 214 and an accelerator 216 (page 9, lines 12-14 and 19-20). The register file 200 provides temporary storage for operands and results (page 9, lines 13-14). The accelerator 216 includes registers for temporary storage of data and control values, and accelerator circuitry for executing specified instructions (page 9, lines 31-33).

A data flow diagram of the operations performed by each accelerator in response to an ACS, or trellis, instruction is shown in FIG. 9 (page 10, lines 4-5). Alpha metrics or beta metrics may be stored in register pairs TRmd and TRnd (page 10, lines 5-6 and 29-30). Gamma transition metrics may be stored in a register Rm (page 10, lines 10-11 and 30-31). The contents of the registers are supplied to subtracting units 250, 254, 260 and 264 and to summing units 252, 256, 258 and 262, as shown in FIG. 9 (page 10, lines 5-13). The outputs of the subtracting units and the summing units are supplied to MAX/TMAX units 270, 272, 274 and 276 (page 10, lines 13-18). The outputs of MAX/TMAX units 270, 272, 274 and 276 are stored in a quad register TRsq (page 10, lines 18-19).

The MAX/TMAX units 270, 272, 274 and 276 each perform one of two functions that may be specified in the trellis instruction. In the MAX function, the maximum of the two inputs is selected and is stored in quad register TRsq. In the TMAX function, the maximum of the two inputs is selected and a correction value is added to the selected maximum value. The sum is stored in quad register TRsq (page 10, lines 20-24).

In a specific example of the add-compare-select operation, a high data word in register pair TRmd (alpha metrics or beta metrics) and a high data word in register Rm (gamma transition metrics) are supplied to subtracting unit 250 (FIG. 9; page 10, lines 5-6 and 10-11). A high data word in register pair TRnd and the high data word in register Rm are supplied to summing unit 258 (FIG. 9; page 10, lines 8-12). The outputs of subtracting unit 250 and summing unit 258 are supplied to MAX/TMAX unit 270 (FIG. 9; page 10, lines 13-15). In the MAX function, the maximum of the two inputs is selected and is stored in quad register TRsq (FIG. 9; page 10, lines 21-22). In the TMAX function, the maximum of the two inputs is selected, a corrected value is

added to the selected maximum value and the sum is stored in quad register TRsq (FIG. 9; page 10, lines 22-24).

In the embodiment of FIG. 9, each accelerator performs two 32-bit butterfly calculations on a trellis in response to a single trellis instruction (page 10, lines 32-33). In another embodiment of the trellis instruction, as shown in FIG. 10, each accelerator performs four 16-bit butterfly calculations in response to a single trellis instruction (page 11, lines 1-2).

An example of software code for calculating alpha metrics and beta metrics of a trellis function is shown in FIG. 11. The code is implemented using ACS instructions which perform the operations shown in FIG. 9 and described above. In FIG. 11, each ACS instruction specifies the calculations for two trellis butterfly calculations as shown in FIG. 9. The ACS instruction is executed in computation blocks 112 and 114 (FIG. 7) to provide a total of four butterfly calculations (page 11, lines 16-21).

The instructions are grouped in pairs in FIG. 11, with a first instruction calculating alpha metrics and beta metrics for the first four states of the trellis at a given time point and the second instruction calculating alpha metrics and beta metrics for the last four states of the trellis at the given time point (page 11, lines 21-24). In the first instruction line of FIG. 11, TR11:8 corresponds to quad register TRsq in FIG. 9. Also in the first instruction line, TR5:4 and TR1:0 correspond to register pair TRmd and register pair TRnd, respectively. The register sR24 corresponds to register Rm in FIG. 9 (page 11, line 32 to page 12, line 1).

In one aspect, the claims on appeal are directed to a method for processing signal values (page 6, lines 1-2; page 11, lines 16-19) comprising: in response to a single trellis instruction (FIG. 11, TR11:8 = ACS(TR5:4, TR1:0, sR24); page 11, lines 16-19; page 10, lines 4-5 and 33-34) that specifies trellis state metrics for a time t_0, based on the signal values, and transition metrics from time t_0 to time t_1, for selected trellis states (FIG. 3; page 6, lines 1-19), a programmable digital signal processor (FIGs. 7 and 8, DSP 110; page 7, lines 15-22; page 9, lines 12-14 and 30-33) executing the steps of: adding a transition metric to a first state metric for

time t_0 to provide a first value (FIG. 9, summing unit 258; page 10, lines 4-15 and 29-31); subtracting the transition metric from a second state metric for time t_0 to provide a second value (FIG. 9, subtracting unit 250; page 10, lines 4-15 and 29-31); for each selected trellis state, comparing the first and second values (FIG. 9, MAX/TMAX unit 270; page 10, lines 20-24); and selecting the maximum of the first and second values for each selected trellis state to provide trellis state metrics for time t_1 (FIG. 9, MAX/TMAX unit 270; page 10, lines 20-24), wherein the adding, subtracting, comparing and selecting operations are executed by the digital signal processor in response to the single trellis instruction (page 10, lines 4-5 and 32-33).

In another aspect, the claims on appeal are directed to a processor for processing signal values (FIGs. 7 and 8, DSP110; page 6, lines 1-2; page 7, lines 15-22; page 9, lines 12-14 and 30-33; page 11, lines 16-19), comprising: a memory (FIG. 7, memory 116; page 7, lines 16-17 and 26-27; page 8, lines 23-26) for storing instructions and operands for digital signal computations; a program sequencer (FIG. 7, control block 124; page 7, lines 16-17) for generating instruction addresses for fetching selected ones of said instructions from said memory; and a computation block (FIG.7, computation blocks 112, 114; page 7, lines 16-22) comprising a register file (FIG. 8, register file 200; page 9, lines 12-14) for temporary storage of operands and results and an accelerator (FIG. 8, accelerator 216; page 9, lines 19-20 and 30-33) for executing a trellis instruction (FIG. 11, TR11:8 = ACS(TR5:4, TR1:0, sR24); page 11, lines 16-19; page 10, lines 4-5 and 33-34) that specifies trellis state metrics for a time t_0 and transition metrics from time t_0 to time t_1 (FIG. 3; page 6, lines 1-19), wherein the trellis state metrics are based on the signal values, said accelerator comprising an adder for adding (FIG. 9, summing unit 258; page 10, lines 4-15 and 29-31) a transition metric to a first state metric for time t_0 to provide a first value, an adder for subtracting (FIG. 9, subtracting unit 250; page 10, lines 4-15 and 29-31) the transition metric from a second state metric for time t_0 to provide a second value, a comparator (FIG. 9, MAX/TMAX unit 270; page 10, lines 20-24) for determining the maximum of the first and second values for each trellis state and a data selector (FIG. 9, MAX/TMAX unit 270; page 10, lines 20-24) for selecting the maximum of the first and second values for selected trellis states to provide trellis state metrics for time t_1, wherein the adders, the comparator and the data selector of

the accelerator are configured to execute the adding, subtracting, comparing and selecting operations in response to a single trellis instruction (page 10, lines 4-5 and 32-33).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1 and 18 stand rejected under 35 U.S.C. §102(b) as anticipated by Amon et al. (U.S. 5,742,621), hereinafter "Amon."

Applicant hereby requests review and reversal of the above grounds of rejection.

There are no rejections under 35 U.S.C.§112.

VII. ARGUMENT

The following argument addresses the issue for appeal.

a. Claims 1 and 18 are Patentable under 35 U.S.C. §102(b) over Amon

Claims 1 and 18 are patentable under 35 U.S.C. §102(b) over Amon because Amon does not each or suggest or all of the limitations of independent claims 1 and 18. This conclusion is supported by the following argument.

Regarding claim 1, Amon does not teach or suggest a method for processing signal values comprising, *in response to a single trellis instruction that specifies trellis state metrics for a time t_0, based on the signal values, and transition metrics from a t_0 to time t_1, for selected trellis states*, a programmable digital signal processor executing the steps of: adding a transition metric to a first state metric for time t_0 to provide a first value, subtracting the transition metric from a second state metric for time t_0 to provide a second value, for each selected trellis state, comparing the first and second values, and selecting the maximum of the first and second values for each selected trellis state to provide trellis state metrics for time t_1, *wherein the adding, subtracting,*

comparing and selecting operations are executed by the digital signal processor in response to the single trellis instruction, as claimed (emphasis added).

Regarding claim 18, Amon does not teach or suggest a processor for processing signal values, comprising a memory for storing instructions and operands for digital signal computations, a program sequencer for generating instruction addresses for fetching selected ones of said instructions from memory, and a computation block comprising a register file for temporary storage of operands and results and *an accelerator for executing a trellis instruction that specifies trellis state metrics for a time t_0 and transition metrics from time t_0 to time t_1, wherein the trellis state metrics are based on the signal values*, said accelerator comprising an adder for adding a transition metric to a first state metric for time t_0 to provide a first value, an adder for subtracting the transition metric from a second state metric for time t_0 to provide a second value, a comparator for determining the maximum of the first and second values for each trellis state and a data selector for selecting the maximum of the first and second values for selected trellis states to provide trellis state metrics for time t_1, *wherein the adders, the comparator and the data selector of the accelerator are configured to execute the adding, subtracting, comparing and selecting operations in response to a single trellis instruction,* as claimed (emphasis added).

Amon discloses a parallel data structure and a dedicated Vertibi specific shift left instruction for minimizing the number of clock cycles required for decoding a convolutionally encoded signal. The data structure and Vertibi shift left instruction reduce the number of clock cycles required for performing an add-compare-select butterfly operation (Abstract). However, Amon does not disclose or suggest a method or processor for processing signal values *wherein adding, subtracting, comparing and selecting operations are executed in response to a single trellis instruction*, as required by claims 1 and 18. As explained below, Amon describes an add-compare-select butterfly operation that utilizes three instructions to perform adding, subtracting and comparing operations. The trellis instruction of the present invention has an advantage that, after an initial latency, one trellis instruction, and thus one ACS operation, can be completed on each clock cycle of the digital signal processor during trellis processing. By contrast, the system of Amon requires three instructions during three clock cycles to complete each and every ACS

operation, even if a pipelined architecture is utilized. When received data is being processed in real time, a three-to-one difference in processing time can be very significant and can determine the difference between having and not having sufficient computation speed to process the data.

The Examiner responds to Applicant's arguments that Amon does not teach or suggest a method or processor for processing signal values wherein adding, subtracting, comparing and selecting operations are executed by a processor in response to single trellis instruction as follows:

> "The Examiner disagrees and asserts that Figure 3 in Amon is an algorithm for carrying out Add/Compare/Select functions in ALU 54 of Figures 1 and 2. The algorithm in Figure 3 begins with a single START instruction and since the algorithm in Figure 3 is an algorithm for carrying out Add/Compare/Select functions over a Trellis, the single START instruction that commences the algorithm in Figure 3 is single Trellis instruction" (Office Action mailed May 2, 2007, page 3)

Applicant acknowledges that FIG. 3 of Amon illustrates a flow chart of an ACS butterfly operation (Col. 6, lines 44-45). However, Applicant does not agree that the algorithm in FIG. 3 of Amon begins with a START "instruction" or that FIG. 3 in any way teaches a single trellis instruction as claimed. In fact, FIG. 3 of Amon does not illustrate any computer instructions. FIG. 3 of Amon is a flow chart that shows steps of an algorithm, not a program listing that shows computer instructions. The START block in FIG. 3 of Amon is a flow chart representation of the starting point of the algorithm and does not represent a computer instruction. Assembly code, including instructions for implementing the ACS butterfly operation, is illustrated in FIGs. 4 and 5 of Amon (Col. 9, lines 24-27 and 37-41). Thus, Applicant contends that FIG. 3 of Amon does not disclose or suggest that adding, subtracting, comparing and selecting operations are executed in response to a single trellis instruction, as required by Applicant's claims 1 and 18.

FIG. 4 of Amon illustrates a first embodiment of assembly code for implementing the ACS butterfly (Col. 9, lines 24-36). As shown in FIG. 4, an add instruction (ADD y1, a) and a

load instruction (l:(r5)-n5, b) appear in the second line of the main ACS loop; a subtract instruction (SUB y1, b) appears in the third line of the main ACS loop; and compare (MAX a, b) and refetch (l:(r5)+n5, a) instructions appear in the fourth line of the main ACS loop. According to convention, assembly code is written with parallel operations in the same line and sequential operations in sequential lines. Thus, Amon implements the ACS operation with add, subtract and compare instructions, which are three separate and distinct instructions that are executed sequentially. FIG. 4 of Amon teaches that three instructions are utilized to perform the add, subtract and compare operations. Thus, FIG. 4 of Amon does not teach or suggest that the adding, subtracting, comparing and selecting operations are executed in response to a single trellis instruction, as required by Applicant's claims 1 and 18.

FIG. 5 of Amon illustrates a second embodiment of assembly code for implementing the ACS butterfly (Col. 9, lines 37-49). Separate add, subtract and compare instructions appear on second, third and fourth lines, respectively, of the main ACS loop. FIG. 5 of Amon teaches that three instructions are utilized to perform add, subtract and compare operations. Thus, FIG. 5 of Amon does not teach or suggest that the adding, subtracting, comparing and selecting operations are executed in response to a single trellis instruction, as required by Applicant's claims 1 and 18.

In rejecting claims 1 and 18 under 35 U.S.C. §102(b) as anticipated by Amon, the Examiner refers to the non-final action of December 19, 2005 for a detailed explanation. In the Office Action of December 19, 2005, the Examiner asserts:

> "Amon teaches that the adding, subtracting, comparing and selecting operations are carried out in Figure 3 in response to a start command comprising trellis states PM1 and PM2 and transition metrics BM...to initiate the adding, subtracting, comparing and selecting operations. Such a start command is a Trellis instruction since it specifies trellis states PM1 and PM2 and transition metrics BM" (Office Action mailed December 19, 2005, page 3)

Applicant contends that Amon contains no such disclosure. FIG. 3 of Amon shows a START block at the beginning of the flow chart. Entirely separate from the START

block are notes which identify the variables BM, PM1, PM2, T1 and T2. This is very different from a start command or a trellis instruction that specifies trellis states and transition metrics. It should be apparent that Amon does not disclose a start command that specifies trellis states and transition metrics, and does not disclose a start command that is a trellis instruction as claimed. Instead, FIG. 3 of Amon shows a flow chart having a START block and separate notes.

The Final Office Action mailed May 2, 2007, on page 3, contains a paragraph that states the Examiner disagrees with the Applicant and maintains all rejections of claims 1-6, 18, 25 and 26. The same paragraph then cites several references "as applied in the last Office Action, filed 12/19/2005". This paragraph appears to be a copy of a paragraph on page 5 of the Office Action mailed May 23, 2006. However, the listed prior art was not applied against the claims in the Office Action of May 23, 2006 or in the Office Action of December 19, 2005. It is submitted that this paragraph is erroneous and therefore is not addressed substantively in this Appeal Brief.

In summary, Amon contains no disclosure or suggestion of a method for processing signal values wherein adding, subtracting, comparing and selecting operations are executed in response to a single trellis instruction, as required by claim 1.

Claims 2-6 depend from claim 1 and are patentable over the cited references for at least the same reasons as claim 1.

Further, Amon contains no disclosure or suggestion of a processor for processing signal values including a computation block comprising a register file and an accelerator for executing a trellis instruction, wherein adders, a comparator and data selector of the accelerator are configured to execute adding, subtracting, comparing and selecting operations in response to a single trellis instruction, as required by claim 18.

Claims 25 and 26 depend from claim 18 and are patentable over the cited references for at least the same reasons as claim 18.

b. Conclusion

For the foregoing reasons, claims 1-6, 18, 25 and 26 are clearly and patentably distinguished over Amon. Accordingly, Applicant respectfully requests reversal of the rejection of claims 1-6, 18, 25 and 26.

VIII. CLAIMS

A copy of the claims involved in the present appeal is attached hereto as Appendix A.

Dated: October 17, 2007

Respectfully submitted,

By William R. McClellan

William R. McClellan
Registration No.: 29,409
WOLF, GREENFIELD & SACKS, P.C.
Federal Reserve Plaza
600 Atlantic Avenue
Boston, Massachusetts 02210-2206
(617) 646-8000

APPENDIX A

Claims Involved in the Appeal of Application Serial No. 09/923,225

1. A method for processing signal values comprising:

in response to a single trellis instruction that specifies trellis state metrics for a time t_0, based on the signal values, and transition metrics from time t_0 to time t_1, for selected trellis states, a programmable digital signal processor executing the steps of:

adding a transition metric to a first state metric for time t_0 to provide a first value;

subtracting the transition metric from a second state metric for time t_0 to provide a second value;

for each selected trellis state, comparing the first and second values; and

selecting the maximum of the first and second values for each selected trellis state to provide trellis state metrics for time t_1, wherein the adding, subtracting, comparing and selecting operations are executed by the digital signal processor in response to the single trellis instruction.

2. A method as defined in claim 1, further comprising the step of, for each selected trellis state, adding to the maximum value a correction factor that is a function of the first and second values.

3. A method as defined in claim 2, wherein the step of adding a correction factor comprises accessing a lookup table containing correction factors.

4. A method as defined in claim 1, wherein the trellis instruction implements a forward trellis function for calculating α trellis state metrics.

5. A method as defined in claim 1, wherein the trellis instruction implements a reverse trellis function for calculating β trellis state metrics.

6. A method as defined in claim 1, wherein the trellis instruction simultaneously implements a forward trellis function for calculating α trellis state metrics and a reverse trellis function for calculating β trellis state metrics, using a single instruction, multiple data approach.

7.-17. (Canceled)

18. A processor for processing signal values, comprising:

a memory for storing instructions and operands for digital signal computations;

a program sequencer for generating instruction addresses for fetching selected ones of said instructions from said memory; and

a computation block comprising a register file for temporary storage of operands and results and an accelerator for executing a trellis instruction that specifies trellis state metrics for a time t_0 and transition metrics from time t_0 to time t_1, wherein the trellis state metrics are based on the signal values, said accelerator comprising an adder for adding a transition metric to a first state metric for time t_0 to provide a first value, an adder for subtracting the transition metric from a second state metric for time t_0 to provide a second value, a comparator for determining the maximum of the first and second values for each trellis state and a data selector for selecting the maximum of the first and second values for selected trellis states to provide trellis state metrics for time t_1, wherein the adders, the comparator and the data selector of the accelerator are configured to execute the adding, subtracting, comparing and selecting operations in response to a single trellis instruction.

19.-24. (Canceled)

25. A processor as defined in claim 18, wherein the accelerator includes an additional adder to add to the maximum of the first and second values a correction factor that is a function of the first and second values.

26. A processor as defined in claim 25, wherein the accelerator further comprises a lookup table containing correction factors.

APPENDIX B

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or entered by or relied upon by the Examiner is being submitted.

APPENDIX C

No related proceedings are referenced in II. above, hence copies of decisions in related proceedings are not provided.